# Symmetry, Equilibria, and Robustness in Common-Payoff Games

Scott Emmons
UC Berkeley
Berkeley, CA
emmons@berkeley.edu

Caspar Oesterheld
Duke University
Durham, NC
caspar.oesterheld@duke.edu

Andrew Critch
UC Berkeley
Berkeley, CA
critch@eecs.berkeley.edu

Vince Conitzer
Duke University
Durham, NC
conitzer@cs.duke.edu

Stuart Russell
UC Berkeley
Berkeley, CA
russell@berkeley.edu

## ABSTRACT

Although it has been known since the 1970s that a *globally* optimal strategy profile in a common-payoff game is a Nash equilibrium, global optimality is a strict requirement that limits the result's applicability. In this work, we show that any *locally* optimal symmetric strategy profile is also a (global) Nash equilibrium. Applied to machine learning, our result provides a global guarantee for any gradient method that finds a local optimum in symmetric strategy space. Furthermore, we show that this result is robust to perturbations to the common payoff and to the local optimum. While these results indicate stability to *unilateral* deviation, we nevertheless identify broad classes of games where mixed local optima are unstable under *joint*, asymmetric deviations. We analyze the prevalence of instability by running learning algorithms in a suite of symmetric games, and we conclude with results on the complexity of computing game symmetries.

## KEYWORDS

common-payoff games, coordination, collaboration, Nash equilibria, symmetry, symmetric games, robustness

**ACM Reference Format:**
Scott Emmons, Caspar Oesterheld, Andrew Critch, Vince Conitzer, and Stuart Russell. 2021. Symmetry, Equilibria, and Robustness in Common-Payoff Games. In *Appears at the 3rd Games, Agents, and Incentives Workshop (GAIW 2021). Held as part of the Workshops at the 20th International Conference on Autonomous Agents and Multiagent Systems., London, UK, May 2021*, IFAAMAS, 17 pages.

## 1 INTRODUCTION

We consider *common-payoff* games (also known as *identical interest* games [38]), in which the payoff to all players is always the same. [1] Such games model a wide range of situations involving cooperative action towards a common goal. Under the heading of *team theory*, they form an important branch of economics [19, 20]. In AI, the common-payoff assumption holds in *Dec-POMDPs* [25], where multiple agents operate independently according to policies designed centrally to achieve a common objective. Many applications of *multiagent reinforcement learning* also assume a common payoff [7, 8, 12]. Finally, in *assistance games* [31] (also known as cooperative inverse reinforcement learning or CIRL games [13]), which include at least one human and one or more "robots," it is assumed that the robots' payoffs are exactly the human's payoff, even if the robots do not know what it is.

Common-payoff games lead naturally to considerations of symmetry in game structure—for example, the assumption that two players' actions produce the same effect on the common payoff. Indeed, von Neumann and Morgenstern [41] and Nash [23] introduced fairly general group-theoretic notions of symmetry, which we adopt and explain in Section 2. More recent work has analyzed *narrower* notions of symmetry [22, 30, 39]. For example, Daskalakis and Papadimitriou [5] study "anonymous games" and show that anonymity substantially reduces the complexity of finding solutions. Finally, Ham [14] generalizes the player-based notion of symmetry to include further symmetries revealed by renamings of actions. We conjecture our results extend to this more general case, at some cost in notational complexity, but we leave this to future work.

In games exhibiting symmetry, it is then reasonable to consider symmetry in players' strategies. (Section 2 defines this in a precise sense.) For example, in team theory, it is common to develop a strategy that can be implemented by every employee in a given category and leads to high payoff for the company. (Notice that this does not lead to identical *behavior*, because strategies are state-dependent.) In civic contexts, symmetry commonly arises through notions of fairness and justice. In treaty negotiations and legislation that mandates how parties behave, for example, there is often a constraint that all parties be treated equally. In DecPOMDPs, an offline solution search may consider only symmetric strategies for identical agents as a way of reducing the search space. In common-payoff multiagent reinforcement learning, each agent may collect percepts and rewards independently, but the reinforcement learning updates can be pooled to learn a single parameterized policy that all agents share.

Common-payoff and symmetric games have a number of desirable properties that may simplify the search for solutions. For the

[1] This condition is relaxed in *(weighted) potential games* where the players' payoffs need only imply the same ordering of outcomes [33]; (weighted) potential games are best-response equivalent to common-payoff games [15, 26].


*Appears at the 3rd Games, Agents, and Incentives Workshop (GAIW 2021). Held as part of the Workshops at the 20th International Conference on Autonomous Agents and Multiagent Systems., Aziz, Ceppi, Dickerson, Hosseini, Lev, Mattei, McElfresh, Zick (chairs), May 2021, London, UK*. 

| Ali \ Bo | L | W |
|---|---|---|
| L | 1 | 2 |
| W | 0 | 1 |

**(a) Bo cannot do laundry**

| Ali \ Bo | L | W |
|---|---|---|
| L | 1 | 2 |
| W | 2 | 1 |

**(b) Bo learns to do laundry**

| Ali \ Bo | L | W |
|---|---|---|
| L | 3 | 2 |
| W | 2 | 3 |

**(c) Joy in working together**

**Table 1: Three versions of the laundry/washing up game. Solutions are described in the text.**

purposes of this paper, we consider Nash equilibria—strategy profiles for all players from which no individual player has an incentive to deviate—as a reasonable solution concept. For example, Marschak and Radner [20] make the obvious point that a globally optimal (possibly asymmetric) strategy profile—one that achieves the highest common payoff—is necessarily a Nash equilibrium. Moreover, it can be found in time linear in the size of the payoff matrix.

Another solution concept often used in multiagent RL and differential (i.e., continuous-action-space) games is that of a *locally optimal* strategy profile—roughly speaking, a strategy profile from which no player has an incentive to slightly deviate. Obviously, a locally optimal profile may not be a Nash equilibrium, as a player may still have an incentive to deviate to some more distant point in strategy space. Nonetheless, local optima, sometimes called *local Nash equilibria*—are important. For example, Ratliff et al. [29] argue that a local Nash equilibrium may still be stable in a practical sense if agents are computationally unable to find a better strategy. Similarly, gradient-based game solvers and multiagent RL algorithms may converge to local optima.

Our first main result, informally stated, is that *in a symmetric, common-payoff game, every local optimum in symmetric strategies is a (global) Nash equilibrium.* Section 3 states the result more precisely and gives an example illustrating its generality.[2] Despite many decades of research on symmetric, common-payoff games, the result appears to be novel and perhaps useful. There are some echoes of the result in the literature on single-agent decision making [4, 27, 32], which can be connected to symmetric solutions of common-payoff games by treating all players jointly as a single agent, but our result appears more general than published results. The proof we give of our result contains elements similar to the proof (of a related but different result) in Taylor [34].

To gain some intuition for these concepts and claims, let us consider a situation in which two children, Ali and Bo, have to do some housework—specifically, laundry ($L$) and washing up ($W$). Here, the "common payoff," if any, is to the parents. It is evident that a symmetric strategy profile—both doing the laundry or both doing the washing up—is not ideal, because the other task will not get done.

The first version of the game, whose payoffs $U$ are shown in Table 1a, is asymmetric: while Ali is competent at both tasks, Bo does not know how to do the laundry properly and will ruin the clothes. Here, as Marschak and Radner pointed out, the strategy profile $(L, W)$ is both globally optimal and a Nash equilibrium. If we posit a mixed (randomized) strategy profile in which Ali and Bo have laundry probabilities $p$ and $q$ respectively, the gradients $\partial U/\partial p$ and $\partial U/\partial q$ are $+1$ and $-1$, driving the solution towards $(L, W)$.

[2]Complete proofs for all of our results are in the appendices.

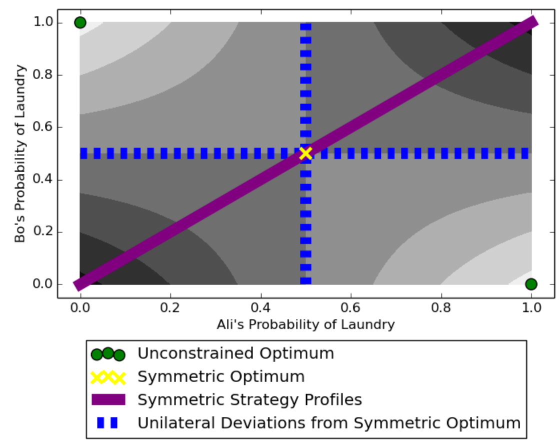


**Figure 1: The strategy profile landscape of the symmetric laundry game (Figure 1b). Although the symmetric optimum has lower expected utility than the unrestricted optima, total symmetry of the game implies that the symmetric optimum is a Nash equilibrium; this is a special case of Theorem 3.2.**

In the second version of the game (Table 1b), Ali has taught Bo how to do the laundry, and symmetry is restored. The pure profiles $(L, W)$ and $(W, L)$ are (asymmetric) globally optimal solutions and hence Nash equilibria. Figure 1 shows the entire payoff landscape as a function of $p$ and $q$: looking just at symmetric strategy profiles, it turns out that there is a local optimum at $p = q = 0.5$, i.e., where Ali and Bo toss fair coins to decide what to do. Although the expected payoff of this solution is lower than that of the asymmetric optima, the local optimum is, nonetheless, a Nash equilibrium. All unilateral deviations from the symmetric local optimum result in the same expected payoff because if one child is tossing a coin, the other child can do nothing to improve the final outcome.

In the third version of the game (Table 1c), the parents derive greater payoff from watching their children working happily together on a single task than they do from getting both tasks done. In this case, there is again a Nash equilibrium at $p = q = 0.5$, but it is a local minimum rather than a local maximum in symmetric strategy space. Thus, not all symmetric Nash equilibria are symmetric local optima; this is because Nash equilibria depend on *unilateral* deviations, whereas symmetric local optima depend on *joint* deviations that maintain symmetry.

In the second half of the paper, we turn to the issue of *robustness* of symmetric solutions. In practice, a variety of factors can lead to modelling errors and approximate solutions, which motivates us to consider perturbations in payoffs and strategy profiles. Making general arguments about Nash equilibria, we show that our first main result is robust in the sense that it degrades linearly under $\epsilon$-magnitude perturbations into $k\epsilon$-Nash equilibria (for some game-dependent constant $k$).

*Stability* turns out to be a thornier issue. Instability, if not handled carefully, might lead to major coordination failures in practice [3]. While it is already known that local strict optima in a totally symmetric team game attain one type of stability, the issue is complex because there are several ways of enforcing (or not enforcing) strict symmetries in payoffs and strategies [22]. Our final results focus on the stability of agents making possibly-asymmetric updates from a symmetric solution. We prove for a non-degenerate class of games that local optima in symmetric strategy space fail to be local optima in asymmetric strategy space if and only if at least one player is mixing, and we experimentally quantify how often mixing occurs for learning algorithms in the GAMUT suite of games [24].

## 2 PRELIMINARIES: GAMES AND SYMMETRIES

### 2.1 Normal-form games

Throughout, we consider *normal-form games* $\mathcal{G} = (N, A, u)$ defined by a finite set $N$ with $|N| = n$ players, a finite set of action profiles $A = A_1 \times A_2 \times \ldots \times A_n$ with $A_i$ specifying the actions available to player $i$, and the utility function $u = (u_1, u_2, \ldots, u_n)$ with $u_i : A \to \mathbb{R}$ giving the utility for each player $i$ [33]. We call $\mathcal{G}$ *common-payoff* if $u_i(a) = u_j(a)$ for all action profiles $a \in A$ and all players $i, j$. In common-payoff games we may omit the player subscript $i$ from utility functions.

We model each player as employing a (mixed) *strategy* $s_i \in \Delta(A_i)$, a probability distribution over actions. We denote the support of the probability distribution $s_i$ by $\text{supp}(s_i)$. Given a (mixed) *strategy profile* $s = (s_1, s_2, \ldots, s_n)$ that specifies a strategy for each player, player $i$'s expected utility is $EU_i(s) = \sum_{a\in A} u_i(a) \prod_{j=1}^{n} s_j(a_j)$. If a strategy $s_i$ for player $i$ maximizes expected utility given the strategies $s_{-i}$ of all the other players, i.e., if $s_i \in \text{argmax}_{s_i' \in \Delta(A_i)} EU_i(s_i', s_{-i})$, we call $s_i$ a *best response* to $s_{-i}$. If each strategy $s_i$ in a strategy profile $s$ is a best response to $s_{-i}$, we call $s$ a *Nash equilibrium.* A Nash equilibrium $s$ is *strict* if every $s_i$ is the unique best response to $s_{-i}$.

Note that, while we have chosen to use the normal-form game representation for simplicity, normal-form games are highly expressive. Normal-form games can represent mixed strategies in all finite games, including games with sequential actions, stochastic transitions, and partial observation such as imperfect-information extensive form games with perfect recall, Markov games, and Dec-POMDPs. To represent a sequential game in normal form, one simply lets each normal-form action be a complete strategy (contingency plan) accounting for every potential game decision.

### 2.2 Symmetry in game structure

Our notion of symmetry in game structure is built upon von Neumann and Morgenstern [41]'s and borrows notation from Plan [28]. The basic building block is a *symmetry of a game*:

*Definition 2.1.* Call a permutation of player indices $\rho : \{1, 2, ..., n\} \to \{1, 2, ..., n\}$ a *symmetry of a game* $\mathcal{G}$ if, for all strategy profiles $(s_1, s_2, ..., s_n)$, permuting the strategy profile permutes the expected payoffs:

$$EU_{\rho(i)}((s_1, s_2, ..., s_n)) = EU_i((s_{\rho(1)}, s_{\rho(2)}, ..., s_{\rho(n)})),\ \forall i.$$

Note that, when we speak of a symmetry of a game, we implicitly assume $A_i = A_j$ for all $i, j$ with $\rho(i) = j$ so that permuting the strategy profile is well-defined. [3]

We characterize the symmetric structure of a game by its set of game symmetries:

*Definition 2.2.* Denote the set of all symmetries of a game $\mathcal{G}$ by: $\Gamma(\mathcal{G}) = \{\rho : \{1, 2, ..., n\} \to \{1, 2, ..., n\} \text{ a symmetry of } \mathcal{G}\}$.

A spectrum of game symmetries is possible. On one end of the spectrum, the identity permutation might be the only symmetry for a given game. On the other end of the spectrum, all possible permutations might be symmetries for a given game. Following the terminology of von Neumann and Morgenstern [41], we call the former case *totally unsymmetric* and the latter case *totally symmetric*:

*Definition 2.3.* If $\Gamma(\mathcal{G}) = S_n$, the full symmetric group, we call the game $\Gamma(\mathcal{G})$ *totally symmetric.* If $\Gamma(\mathcal{G})$ contains only the identity permutation, we call the game *totally unsymmetric.*

Let $\mathcal{P} \subseteq \Gamma(\mathcal{G})$ be any subset of the game symmetries. Because $\Gamma(\mathcal{G})$ is closed under composition, we can repeatedly apply permutations in $\mathcal{P}$ to yield a group of game symmetries $\langle\mathcal{P}\rangle$:

*Definition 2.4.* Let $\mathcal{P} \subseteq \Gamma(\mathcal{G})$ be a subset of the game symmetries. The group *generated* by $\mathcal{P}$, denoted $\langle\mathcal{P}\rangle$, is the set of all permutations that can result from (possibly repeated) composition of permutations in $\mathcal{P}$: $\langle\mathcal{P}\rangle = \{\rho_1 \circ \rho_2 \circ \ldots \circ \rho_m \mid m \in \mathbb{N}, \rho_1, \rho_2, \ldots, \rho_m \in \mathcal{P}\}$.

Group theory tells us that $\langle\mathcal{P}\rangle$ defines a closed binary operation (permutation composition) including an identity and inverse maps, and $\langle\mathcal{P}\rangle$ is the closure of $\mathcal{P}$ under function composition.

With a subset of game symmetries $\mathcal{P} \subseteq \Gamma(\mathcal{G})$ in hand, we can use the permutations in $\mathcal{P}$ to carry one player index to another. For each player $i$, we give a name to the set of player indices to which permutations in $\mathcal{P}$ can carry $i$: we call it player $i$'s *orbit.*

*Definition 2.5.* Let $\mathcal{P} \subseteq \Gamma(\mathcal{G})$ be a subset of the game symmetries $\Gamma(\mathcal{G})$. The *orbit* of player $i$ under $\mathcal{P}$ is the set of all other player indices that $\langle\mathcal{P}\rangle$ can assign to $i$: $\mathcal{P}(i) = \{\rho(i) \mid \rho \in \langle\mathcal{P}\rangle\}$.

In fact, it is a standard result from group theory that the orbits of a group action on a set partition the set's elements, which leads to the following proposition:

Proposition 2.6. *Let $\mathcal{P} \subseteq \Gamma(\mathcal{G})$. The orbits of $\mathcal{P}$ partition the game's players.*

By Proposition 2.6, each $\mathcal{P} \subseteq \Gamma(\mathcal{G})$ yields an equivalence relation among the players. To gain intuition for this equivalence relation, consider two extreme cases. In a totally unsymmetric game, $\Gamma(\mathcal{G})$ contains only the identity permutation, in which case each player

[3] We make this choice to ease notational burden, but we conjecture that our results can be generalized to allow for mappings between actions [14], which we leave for future work.

is in its own orbit of $\Gamma(\mathcal{G})$; the equivalence relation induced by the orbit partition shows that no players are equivalent. In a totally symmetric game, by contrast, every permutation is an element of $\Gamma(\mathcal{G})$, i.e., $\Gamma(\mathcal{G}) = S_n$, the full symmetric group; now, all the players share the same orbit of $\Gamma(\mathcal{G})$, and the equivalence relation induced by the orbit partition shows that all the players are equivalent.

We leverage the orbit structure of an arbitrary $\mathcal{P} \subseteq \Gamma(\mathcal{G})$ to define an equivalence relation among players because it adapts to however much or little symmetry is present in the game. Between the extreme cases of no symmetry ($n$ orbits) and total symmetry (1 orbit) mentioned above, there could be any intermediate number of orbits of $\mathcal{P}$. Furthermore, two players can share an orbit of $\mathcal{P}$ even if those two players cannot be arbitrarily swapped. In Example 3.3, all the players can be rotated in a circle, so all the players share an orbit of $\mathcal{P} = \Gamma(\mathcal{G})$ even though the game does not admit arbitrary swapping of players.

## 2.3 Symmetry in strategy profiles

Having formalized a symmetry of a game in the preceding section, we follow Nash [23] and define symmetry in strategy profiles with respect to symmetry in game structure:

*Definition 2.7.* Let $\mathcal{P} \subseteq \Gamma(\mathcal{G})$ be a subset of the game symmetries $\Gamma(\mathcal{G})$. We call a strategy profile $s = (s_1, s_2, ..., s_n)$ *$\mathcal{P}$-invariant* if $(s_1, s_2, ..., s_n) = (s_{\rho(1)}, s_{\rho(2)}, ..., s_{\rho(n)})$ for all $\rho \in \langle \mathcal{P} \rangle$.

The equivalence relation among players induced by the orbit structure of $\mathcal{P}$ is fundamental to our definition of symmetry in strategy profiles by the following proposition:

Proposition 2.8. *A strategy profile $s = (s_1, s_2, ..., s_n)$ is $\mathcal{P}$-invariant if and only if $s_i = s_j$ for each pair of players $i$ and $j$ with $\mathcal{P}(i) = \mathcal{P}(j)$.*

To state Proposition 2.8 another way, a strategy profile is $\mathcal{P}$-invariant if all pairs of players $i$ and $j$ that are equivalent under the orbits of $\mathcal{P}$ play the same strategy.

# 3 LOCAL SYMMETRIC OPTIMA ARE (GLOBAL) NASH EQUILIBRIA

After the formal definitions of symmetry in the previous section, we are almost ready to formally state the first of our three main results. The only remaining definition is that of a local symmetric optimum:

*Definition 3.1.* Call $s$ a *locally optimal $\mathcal{P}$-invariant* strategy profile of a common-payoff game if: (i) $s$ is $\mathcal{P}$-invariant, and (ii) for some $\epsilon > 0$, no $\mathcal{P}$-invariant strategy $s'$ with $EU(s') > EU(s)$ can be formed by adding or subtracting at most $\epsilon$ to the probability of taking any given action $a_i \in A_i$. If, furthermore, condition (ii) holds *for all $\epsilon > 0$*, we call $s$ a *globally optimal $\mathcal{P}$-invariant* strategy profile or simply an *optimal $\mathcal{P}$-invariant* strategy profile.

Now we can formally state our first main theorem, that local symmetric optima are (global) Nash equilibria:

Theorem 3.2. *Let $\mathcal{G}$ be a common-payoff normal-form game, and let $\mathcal{P} \subseteq \Gamma(\mathcal{G})$ be a subset of the game symmetries $\Gamma(\mathcal{G})$. Any locally optimal $\mathcal{P}$-invariant strategy profile is a Nash equilibrium.*

Proof. We provide a sketch here and full details in Appendix A. Suppose, for the sake of contradiction, that an individual player $i$ could beneficially deviate to action $a_i$ (if a beneficial deviation exists, then there is one to a pure strategy). Then, consider instead a collective change to a symmetric strategy profile in which all the players in $i$'s orbit shift very slightly more probability to $a_i$. By making the amount of probability shifted ever smaller, the probability that this change affects exactly one agent's realized action (making it $a_i$ when it would not have been before) can be arbitrarily larger than the probability that it affects multiple agents' realized actions. Moreover, if this causes exactly one agent's realized action to change, this must be in expectation beneficial, since the original unilateral deviation was in expectation beneficial. Hence, the original strategy profile cannot have been locally optimal. □

## 3.1 Example illustrating general symmetry

Here, we give an example that shows how Theorem 3.2 is more general than the case of total symmetry. The example illustrates the existence of rotational symmetry without total symmetry, and it illustrates how picking different $\mathcal{P} \subseteq \Gamma(\mathcal{G})$ leads to different optimal $\mathcal{P}$-invariant strategies and thus different $\mathcal{P}$-invariant Nash equilibria by Theorem 3.2.

*Example 3.3.* There are four radio stations positioned in a square. We number these 1,2,3,4 clockwise, such that, e.g., 1 neighbors 4 and 2. There is also a neighborhood of people at each vertex of the square. The people can tune in to the radio station at their vertex of the square and to the radio stations at adjacent vertices of the square, but they cannot tune in to the station at the opposite vertex.

The game has each radio station choose what to broadcast. For simplicity, suppose each radio station can broadcast the weather or music. The common payoff of the game is the sum of the utilities of the four neighborhoods. For each neighborhood, if the neighborhood cannot tune in to the weather, the payoff for that neighborhood is 0. If the neighborhood can only tune in to the weather, the payoff is 1, and if the neighborhood can tune in to both weather and music, the neighborhood's payoff is 2.

The symmetries of the game $\Gamma(\mathcal{G})$ include the set of permutations generated by rotating the radio stations once clockwise. In standard notation for permutations, $\{(1, 2, 3, 4), (2, 3, 4, 1), (3, 4, 1, 2), (4, 1, 2, 3)\} \subset \Gamma(\mathcal{G})$.

First, consider applying the theorem to $\mathcal{P} = \Gamma(\mathcal{G})$. In this case, the constraint of $\mathcal{P}$-invariance requires all the radio stations to play the same strategy because all stations are in the same orbit. As we show in Appendix B, the optimal $\mathcal{P}$-invariant strategy then is for each station to broadcast music with probability $\sqrt{2}-1$. Theorem 3.2 tells us that this optimal $\mathcal{P}$-invariant strategy profile is a Nash equilibrium. Appendix B also shows how to verify this without the use of Theorem 3.2.

Second, consider applying the theorem to the case where $\mathcal{P}$ consists only of the rotation twice clockwise, i.e., the permutation which maps each station onto the station on the opposite vertex of the square. In standard notation for permutations, $\mathcal{P} = \{(3, 4, 1, 2)\}$. Now, the constraint of $\mathcal{P}$-invariance requires radio stations at opposite vertices of the square to play the same strategy. However, neighboring stations can broadcast different programs. The optimal $\mathcal{P}$-invariant strategy is for one pair of opposite-vertex radio stations, e.g., 1 and 3, to broadcast the weather and for the other pair of radio stations, 2 and 4, to broadcast music. While it turns out to be immediate that this optimal $\mathcal{P}$-invariant strategy is a Nash equilibrium because it achieves the globally optimal outcome, we could have applied Theorem 3.2 to know that this optimal $\mathcal{P}$-invariant strategy profile is a Nash equilibrium *even without knowing what the optimal $\mathcal{P}$-invariant strategy was.*

# 4 ROBUSTNESS OF THE MAIN RESULT TO PAYOFF AND STRATEGY PERTURBATIONS

The first type of robustness we consider is robustness to perturbations in the game's payoff function. Formally, we define an $\epsilon$-*perturbation* of a game as follows:

*Definition 4.1.* Let $\mathcal{G}$ be a normal-form game with utility function $\mu$. For some $\epsilon > 0$, we call $\mathcal{G}'$ an $\epsilon$-*perturbation* of $\mathcal{G}$ if $\mathcal{G}'$ has utility function $\mu'$ satisfying $\max_{i \in N, a \in A} |u_i'(a) - u_i(a)| \leq \epsilon$.

There are a variety of reasons why $\epsilon$-perturbations might arise in practice. Our game model may contain errors such as the game not being perfectly symmetric; the players' preferences might drift over time; or we might have used function approximation to learn the game's payoffs. With Proposition 4.2, we note a generic observation about Nash equilibria showing that our main result, Theorem 3.2, is robust in the sense of degrading *linearly* in the payoff perturbation's size:

Proposition 4.2. *Let $\mathcal{G}$ be a common-payoff normal-form game, and let $s^*$ be a locally-optimal $\mathcal{P}$-invariant strategy profile for some subset of game symmetries $\mathcal{P} \subseteq \Gamma(\mathcal{G})$. Suppose $G'$ is an $\epsilon$-perturbation of $\mathcal{G}$. Then $s^*$ is a $2\epsilon$-Nash equilibrium in $\mathcal{G}'$.*

The second type of robustness we consider is robustness to symmetric solutions that are only approximate. For example, we might try to find a symmetric local optimum through an approximate optimization method, or the evolutionary dynamics among players' strategies might lead them to approximate local symmetric optima. Again, a generic result about Nash equilibria shows that the guarantee of Theorem 3.2 degrades linearly in this case:

Theorem 4.3. *Let $\mathcal{G}$ be a common-payoff normal-form game, and let $s^*$ be a locally-optimal $\mathcal{P}$-invariant strategy profile for some subset of game symmetries $\mathcal{P} \subseteq \Gamma(\mathcal{G})$. Suppose $s$ is a strategy profile with total variation distance $TV(s, s^*) \leq \delta$. Then $s$ is an $\epsilon$-Nash equilibrium with $\epsilon = 4\delta \max_{i \in N, a \in A} |u_i(a)|$.*

By Theorem 4.3, we have a robustness guarantee in terms of the total variation distance between an approximate local symmetric optimum and a true local symmetric optimum. Without much difficulty, we can also convert this into a robustness guarantee in terms of the Kullback-Leibler divergence:

Corollary 4.4. *Let $\mathcal{G}$ be a common-payoff normal-form game, and let $s^*$ be a locally-optimal $\mathcal{P}$-invariant strategy profile for some subset of game symmetries $\mathcal{P} \subseteq \Gamma(\mathcal{G})$. Suppose $s$ is a strategy profile with Kullback-Leibler divergence satisfying $D_{KL}(s||s^*) \leq \nu$ or $D_{KL}(s^*||s) \leq \nu$. Then $s$ is an $\epsilon$-Nash equilibrium with $\epsilon = 2\sqrt{2\nu} \max_{i \in N, a \in A} |u_i(a)|$.*

While the results of this section show the robustness of Nash equilibria, we note that Nash equilibria, by definition, consider the possibility of only a single agent deviating; Nash equilibria cannot guarantee stability under dynamics that allow for multiple agents to deviate. In the next section, we investigate when multiple agents might have an incentive to simultaneously deviate by studying the optimality of symmetric strategy profiles in possibly-asymmetric strategy space.

# 5 WHEN ARE LOCAL OPTIMA IN SYMMETRIC STRATEGY SPACE ALSO LOCAL OPTIMA IN POSSIBLY-ASYMMETRIC STRATEGY SPACE?

Our first main theoretical result, Theorem 3.2, applies to locally optimal $\mathcal{P}$-invariant, i.e., symmetric, strategy profiles. This still leaves open the question of how well locally optimal symmetric strategy profiles perform when considered in the broader, possibly-asymmetric strategy space. When are locally optimal $\mathcal{P}$-invariant strategy profiles also locally optimal in possibly-asymmetric strategy space? This question is important in machine learning (ML) applications where users of symmetrically optimal ML systems might be motivated to make modifications to the systems, even for purposes of a common payoff.

To address this issue more precisely, we formally define a *local optimum in possibly-asymmetric strategy space*:

*Definition 5.1.* A strategy profile $s = (s_1, s_2, \ldots, s_n)$ of a common-payoff normal-form game is *locally optimal among possibly-asymmetric strategy profiles*, or, equivalently, a *local optimum in possibly-asymmetric strategy space*, if for some $\epsilon > 0$, no strategy profile $s'$ with $EU(s') > EU(s)$ can be formed by changing $s$ in such a way that the probability of taking any given action $a_i \in A_i$ for any player $i$ changes by at most $\epsilon$.

Definition 5.1 relates to notions of *stability* under dynamics, such as those with perturbations or stochasticity, that allow multiple players to make asymmetric deviations. In particular, if $s$ is not a local maximum in asymmetric strategy space, this means that there is some set of players $C$ and strategy $s_C'$ arbitrarily close to $s$, such that if players $C$ were to play $s_C'$ (by mistake or due to stochasticity), some Player $i \in N - C$ would develop a strict preference over the support of $s_i$. To illustrate this, we return to the laundry/washing up game of the introduction.

*Example 5.2.* Consider again the game of Table 1b. As Figure 1 illustrates, the symmetric optimum is for both Ali and Bo to randomize uniformly between W and L. While this is a Nash equilibrium, it is not a local optimum in possibly-asymmetric strategy space. If one player deviates from uniformly randomizing, the other player develops a strict preference for either $W$ or $L$.

To understand when the phenomenon of Example 5.2 happens in general, we use the following *degeneracy* condition:

*Definition 5.3.* Let $s$ be a Nash equilibrium of a game $\mathcal{G}$:

- If $s$ is deterministic, i.e., if every $s_i$ is a Dirac delta function on some $a_i$, then $s$ is *degenerate* if at least two players $i$ are indifferent between $a_i$ and some other $a_i' \in A_i - \{a_i\}$.
- Otherwise, if $s$ is mixed, then $s$ is *degenerate* if for all players $i$ and all $a_{-i} \subseteq \text{supp}(s_{-i})$, the term $EU_i(a_i, a_{-i})$ is constant across $a_i \in \text{supp}(s_i)$.

Intuitively, our definition says that a deterministic Nash equilibrium is non-degenerate when it is strict or almost strict (allowing the exception of at most one player who may be indifferent over available actions). A mixed Nash equilibrium, on the other hand, is non-degenerate when *mixing matters*. When speaking of a game $\mathcal{G}$, we determine its degeneracy by the degeneracy of its Nash equilibria:

*Definition 5.4.* We call a game $\mathcal{G}$ *degenerate* if it has at least one degenerate Nash equilibrium; otherwise, we call $\mathcal{G}$ *non-degenerate*.

We note that "degnerate" is already an established term in the game-theoretical literature where it is often applied only to two-player games [see, e.g, 42, Definition 3.2]. While similar to the established notion of degeneracy, our definition of degeneracy is stronger, which makes our statements about non-degenerate games more general. If a two-player game $\mathcal{G}$ is non-degenerate in the usual sense from the literature, it is non-degenerate in the sense of Definition 5.3. Moreover, if $\mathcal{G}$ is common-payoff, then for each player $i$, we can define a two-player game played by $i$ and another single player who controls the strategies of $N - \{i\}$. If for all $i$ these two-player games are non-degenerate in the established sense, then $\mathcal{G}$ is non-degenerate in the sense of Definition 5.3.

In non-degenerate games, our next theorem shows that a local symmetric optimum is a local optimum in possibly-asymmetric strategy space if and only if it is deterministic. Formally:

Theorem 5.5. *Let $\mathcal{G}$ be a non-degenerate common-payoff normal-form game, and let $\mathcal{P} \subseteq \Gamma(\mathcal{G})$ be a subset of the game symmetries $\Gamma(\mathcal{G})$. A locally optimal $\mathcal{P}$-invariant strategy profile is locally optimal among possibly-asymmetric strategy profiles if and only if it is deterministic.*

To see why the (non-)degeneracy condition is needed in Theorem 5.5, we provide an example of a degenerate game:

*Example 5.6.* Consider the 3x3 symmetric game with the following payoff matrix:

| | | Player 2 | | |
|---|---|---|---|---|
| | | a | b | c |
| Player 1 | a | 1 | 1 | 1 |
| | b | 1 | -10 | $1+\epsilon$ |
| | c | 1 | $1+\epsilon$ | -10 |

Here, $(a, a)$ is the unique global optimum in symmetric strategy space. By Theorem 3.2, it is therefore also a Nash equilibrium. However, it is a degenerate Nash equilibrium and not locally optimal in asymmetric strategic space. The payoff can be improved by, e.g., Player 1 playing $b$ with small probability (and $a$ otherwise) and Player 2 playing $c$ with small probability (and $a$ otherwise).

The following game illustrates how a global symmetric optimum, even if it is a non-degenerate, deterministic equilibrium, might still not be *globally* optimal in possibly-asymmetric strategy space.

*Example 5.7.* Consider $|N| = 3$ and $A = \{0, 1\}$, let $k$ be the number of players who choose action 1, and let the payoffs be: 0 if $k = 0$, $-1$ if $k = 1$, 1 if $k = 2$, and $-1$ if $k = 3$.

Then the global symmetric optimum is for everyone to play 0. The global asymmetric optimum, on the other hand, is to coordinate to achieve $k = 2$. Hence, the global symmetric optimum is strictly worse than the global asymmetric optimum. Of course, by Theorem 5.5, $(0, 0, 0)$ is still a local optimum of asymmetric strategy space.

# 6 LEARNING SYMMETRIC STRATEGIES IN GAMUT

Theorem 5.5 shows that, in non-degenerate games, a locally optimal symmetric strategy profile is stable in the sense of Section 5 if and only if it is pure. For those concerned about stability, this raises the question: how often are optimal strategies pure, and how often are they mixed?

To answer this question, we present an empirical analysis of learning symmetric strategy profiles in the GAMUT suite of game generators [24]. We are interested both in how centralized optimization algorithms (such as gradient methods) search for symmetric strategies and in how decentralized populations of agents evolve symmetric strategies. To study the former, we run Sequential Least SQuares Programming (SLSQP) [17, 40], a local search method for constrained optimization. To study the latter, we simulate the replicator dynamics [9], an update rule from evolutionary game theory with connections to reinforcement learning [2, 36, 37]. (See Appendix E.3 for details.)

## 6.1 Experimental setup

We ran experiments in all three classes of symmetric GAMUT games: RandomGame, CoordinationGame, and CollaborationGame. Intuitively, a RandomGame draws all payoffs uniformly at random, whereas in a CoordinationGame and a CollaborationGame, the highest payoffs are always for outcomes where all players choose the same action. (See Appendix E.1 details.) Because CoordinationGame and CollaborationGame have such similar game structures, our experimental results in the two games are nearly identical. To avoid redundancy, we only include experimental results for CoordinationGame in this paper.

For each game class, we sweep the parameters of the game from 2 to 5 players and 2 to 5 actions, i.e., with $(|N|, |A_i|) \in \{2, 3, 4, 5\} \times \{2, 3, 4, 5\}$. We sample 100 games at each parameter setting and then attempt to calculate the global symmetric optimum using (i) 10 runs of SLSQP and (ii) 10 runs of the replicator dynamic (each with a different initialization drawn uniformly at random over the simplex), resulting in 10 + 10 = 20 solution attempts per game. Because we do not have ground truth for the globally optimal solution of the game, we instead use the best of our 20 solution attempts, which we call the "best solution."

To apply our previously developed theory to GAMUT games, we observe that RandomGames, CoordinationGames, and CollaborationGames are (almost surely) non-degenerate in the sense of Definition 5.4:

Proposition 6.1. *Drawing a degenerate game is a measure-zero event in RandomGames, CoordinationGames, and CollaborationGames.*

## 6.2 What fraction of symmetric optima are local optima in possibly-asymmetric strategy space?

Here, we try to get a sense for how often symmetric optima are stable in the sense that they are also local optima in possibly-asymmetric strategy space (see Section 5). In Appendix Table 3b, we show in what fraction of games the best solution of our 20 optimization attempts is mixed; by Theorem 5.5, this is the fraction of games whose symmetric optima are not local optima in possibly-asymmetric strategy space. In CoordinationGames, the symmetric optimum is always (by construction) for all players to choose the same action, leading to stability. By contrast, we see that 36% to 60% of RandomGames are *unstable*. We conclude that if real-world games do not have the special structure of CoordinationGames, then instability may be common.

## 6.3 How often do SLSQP and the replicator dynamic find an optimal solution?

As sequential least squares programming and the replicator dynamic are not guaranteed to converge to a global optimum, we test empirically how often each run converges to the best solution of our 20 optimization runs. In Appendix Table 4 / Table 6, we show what fraction of the time any single SLSQP / replicator dynamics run finds the best solution, and in Appendix Table 5 / Table 7, we show what fraction of the time at least 1 of 10 SLSQP / replicator dynamics runs finds the best solution. First, we note that the tables for SLSQP and the replicator dynamics are quite similar, differing by no more than a few percentage points in all cases. So the replicator dynamics, which are used as a model for how populations evolve strategies, can also be used as an effective optimization algorithm. Second, we see that individual runs of each algorithm are up to 93% likely to find the best solution in small RandomGames, but they are less likely (as little as 24% likely) to find the best solution in larger RandomGames and in CoordinationGames. The best of 10 runs, however, finds the best solution $\geq$ 87% of the time, indicating that random algorithm restarts benefit symmetric strategy optimization.

# 7 COMPUTATIONAL COMPLEXITY OF COMPUTING GAME SYMMETRIES AND SYMMETRIC STRATEGIES

## 7.1 Finding symmetries

In some cases, domain knowledge can provide the symmetries of a game. For example, in the laundry game of Table 1b, symmetry arises from a simple observation: it matters only *what* chores get done, not *which* children do the chores. In other cases, however, players may face a potentially symmetric common-payoff game and first have to determine what the symmetries of the game are, e.g., by computing a generating set of the group of symmetries. Call this problem the game automorphism (GA) problem. Can it be solved efficiently?

The complexity of the GA problem depends on how the game is represented. The simplest representation is to give the full table of payoffs. However, the size is then exponential in the number of players. A simple alternative is to only explicitly represent non-zero entries in the payoff table. This way, some games of many players can be represented succinctly. Calling the latter a *sparse* representation and the former a *non-sparse* representation, we obtain the following:

Theorem 7.1. *On a non-sparse game representation, the GA problem can be solved in polynomial time. On a sparse representation, the GA problem is polynomial-time equivalent to the graph isomorphism problem.*

For a general introduction to the graph isomorphism problem, see Grohe and Schweitzer [11]. Notably, the problem is in NP but neither known to be solvable in polynomial time nor known to be NP-hard.

## 7.2 Finding an optimal symmetric strategy profile

Once it is known what the symmetries $\mathcal{P}$ of a given game are, what is the complexity of finding an optimal $\mathcal{P}$-invariant strategy profile? In Appendix G.2, we show that the problem of optimizing symmetric strategies is equivalent to the problem of optimizing polynomials on Cartesian products of unit simplices. However, depending on how the polynomials and games are represented, the reductions may increase the problem instance exponentially. Nevertheless, we can import results from the literature on optimizing polynomials to obtain results such as the following:

Theorem 7.2. *Deciding for a given game $\mathcal{G}$ with symmetries $\mathcal{P}$ and a given number $K$ whether there is a $\mathcal{P}$-invariant profile with expected utility at least $K$ is NP-hard, even for 2-player symmetric games.*

# 8 CONCLUSION

When ML is deployed in the world, it is natural to instantiate multiple agents from the same template. This naturally restricts strategy profiles to *symmetric* ones, and it puts the focus on finding optimal symmetric strategy profiles. This, in turn, raises questions about the properties of such profiles. Would individual agents (or the users they serve) want to deviate from these profiles? Are they robust to small changes in the game or in the executed strategies? Could there be better asymmetric strategy profiles nearby?

Our results yield a mix of good and bad news. Theorems 3.2 and 4.3 are good news, showing that even local optima in symmetric strategy space are (global) Nash equilibria in a robust sense. So, with respect to *unilateral* deviations among team members, symmetric optima are relatively stable strategies. On the other hand, Theorem 5.5 is perhaps bad news, because it shows that a broad class of symmetric local optima are unstable when considering *joint* deviations in asymmetric strategy space (Section 5). Furthermore, our empirical results with learning algorithms in GAMUT suggest that these unstable solutions may not be uncommon in practice (Section 6.2).

# REFERENCES

[1] V. Arvind, Bireswar Das, Johannes Köbler, and Seinosuke Toda. 2010. Colored Hypergraph Isomorphism is Fixed Parameter Tractable. In *Proceedings of the IARCS International Conference on Foundations of Software Technology and Theoretical Computer Science*. Dagstuhl Publishing, 327–337. https://doi.org/10.4230/LIPIcs.FSTTCS.2010.327

[2] Tilman Börgers and Rajiv Sarin. 1997. Learning through reinforcement and replicator dynamics. *Journal of economic theory* 77, 1 (1997), 1–14.

[3] Nick Bostrom, Thomas Douglas, and Anders Sandberg. 2016. The Unilateralist's Curse and the Case for a Principle of Conformity. *Social epistemology* 30, 4 (2016), 350–371.

[4] Rachael Briggs. 2010. Putting a value on Beauty. In *Oxford Studies in Epistemology*. Vol. 3. Oxford University Press, 3–24.

[5] Constantinos Daskalakis and Christos H. Papadimitriou. 2007. Computing Equilibria in Anonymous Games. In *FOCS*.

[6] Etienne de Klerk. 2008. The complexity of optimizing over a simplex, hypercube or sphere: a short survey. *Central European Journal of Operations Research* 16 (2008), 111–125. https://link.springer.com/article/10.1007/s10100-007-0052-9

[7] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in neural information processing systems*. 2137–2145.

[8] Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Thirty-second AAAI conference on artificial intelligence*.

[9] Drew Fudenberg and David K Levine. 1998. *The theory of learning in games*. Vol. 2. MIT press.

[10] Joaquim Gabarró, Alina García, and Maria Serna. 2011. The complexity of game isomorphism. *Theoretical Computer Science* 412, 48 (11 2011), 6675–6695. https://doi.org/10.1016/j.tcs.2011.07.022

[11] Martin Grohe and Pascal Schweitzer. 2020. The Graph Isomorphism Problem. *Commun. ACM* 63, 11 (11 2020), 128–134. https://doi.org/10.1145/3372123

[12] Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. 2017. Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*. Springer, 66–83.

[13] Dylan Hadfield-Menell, Anca D. Dragan, Pieter Abbeel, and Stuart J. Russell. 2017. Cooperative inverse reinforcement learning. In *Advances in Neural Information Processing 29*.

[14] Nicholas Ham. 2013. Notions of Symmetry for Finite Strategic-Form Games. arXiv:1311.4766.

[15] Ramesh Johari. 2007. Fictitious play: Examples and convergence. URL: http://web.stanford.edu/~rjohari/teaching/notes/336_lecture7_2007.pdf.

[16] Matthias Köppe. 2010. *On the Complexity of Nonlinear Mixed-Integer Optimization*. https://arxiv.org/pdf/1006.4895.pdf Also published in https://link.springer.com/chapter/10.1007/978-1-4614-1927-3_19.

[17] Dieter Kraft. 1988. *A Software Package for Sequential Quadratic Programming*. Technical Report DFVLR-FB 88-28. DLR German Aerospace Center – Institute for Flight Mechanics, Koln, Germany.

[18] Eugene M. Luks. 1999. Hypergraph Isomorphism and Structural Equivalence of Boolean Functions. In *STOC '99: Proceedings of the thirty-first annual ACM symposium on Theory of Computing*. Atlanta, GA, USA, 652–658.

[19] Jakob Marschak. 1955. Elements for a theory of teams. *Management Science* 1, 2 (1955), 127–137.

[20] Jacob Marschak and Roy Radner. 1972. *Economic Theory of Teams*. Yale University Press.

[21] Rudolf Mathon. 1979. A note on the graph isomorphism counting problem. *Inform. Process. Lett.* 8, 3 (3 1979), 131–136. https://doi.org/10.1016/0020-0190(79)90004-8

[22] Igal Milchtaich. 2016. *Static stability in symmetric and population games*. Technical Report. https://www.biu.ac.il/soc/ec/wp/2008-04/2008-04.pdf

[23] John Nash. 1951. Non-cooperative games. *Annals of Mathematics* (1951), 286–295.

[24] E Nudelman, J Wortman, Y Shoham, and K Leyton-Brown. 2004. Run the GAMUT: a comprehensive approach to evaluating game-theoretic algorithms. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.* IEEE, 880–887.

[25] Frans A Oliehoek, Christopher Amato, et al. 2016. *A Concise Introduction to Decentralized POMDPs*. Springer.

[26] Asu Ozdaglar. 2010. Learning in Games. URL: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-254-game-theory-with-engineering-applications-spring-2010/lecture-notes/MIT6_254S10_lec11.pdf.

[27] Michele Piccione and Ariel Rubinstein. 1997. On the interpretation of decision problems with imperfect recall. *Games and Economic Behavior* 20, 1 (1997), 3–24.

[28] Asaf Plan. 2017. Symmetric n-player games. Working paper, available at asafplan.com.

[29] Lillan J. Ratliff, Samuel A. Burden, and S. Shankar Sastry. 2016. On the Characterization of Local Nash Equilibria in Continuous Games. *IEEE Trans. Automat. Control* 61, 8 (Aug. 2016), 2301–2307. https://doi.org/10.1109/TAC.2016.2583518

[30] Philip J Reny. 1999. On the existence of pure and mixed strategy Nash equilibria in discontinuous games. *Econometrica* 67, 5 (1999), 1029–1056.

[31] Stuart Russell. 2019. *Human Compatible: Artificial Intelligence and the Problem of Control*. Viking Press.

[32] Wolfgang Schwarz. 2015. Lost memories and useless coins: revisiting the absent-minded driver. *Synthese* 192 (2015), 3011–3036.

[33] Yoav Shoham and Kevin Leyton-Brown. 2008. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press. http://www.masfoundations.org/

[34] Jessica Taylor. 2016. In memoryless Cartesian environments, every UDT policy is a CDT+SIA policy. https://www.alignmentforum.org/posts/5bd75cc58225bf06703751b2/in-memoryless-cartesian-environments-every-udt-policy-is-a-cdt-sia-policy

[35] Alexandre Tsybakov. 2009. *Introduction to Nonparametric Estimation*. Springer.

[36] Karl Tuyls, Dries Heytens, Ann Nowe, and Bernard Manderick. 2003. Extended replicator dynamics as a key to reinforcement learning in multi-agent systems. In *European Conference on Machine Learning*. Springer, 421–431.

[37] Karl Tuyls, Katja Verbeeck, and Tom Lenaerts. 2003. A selection-mutation model for q-learning in multi-agent systems. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. 693–700.

[38] Takashi Ui. 2009. Bayesian potentials and information structures: Team decision problems revisited. *International Journal of Economic Theory* 5, 3 (2009), 271–291.

[39] Steen Vester. 2012. *Symmetric Nash Equilibria*. Ph.D. Dissertation.

[40] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2

[41] John von Neumann and Oskar Morgenstern. 1944. *Theory of Games and Economic Behavior*. Princeton University Press.

[42] Bernhard von Stengel. 2007. Equilibrium Computation for Two-Player Games in Strategic and Extensive Form. In *Algorithmic Game Theory*. Cambridge University Press.

## A PROOFS OF SECTION 3 RESULTS

Theorem 3.2. *Let $\mathcal{G}$ be a common-payoff normal-form game, and let $\mathcal{P} \subseteq \Gamma(\mathcal{G})$ be a subset of the game symmetries $\Gamma(\mathcal{G})$. Any locally optimal $\mathcal{P}$-invariant strategy profile is a Nash equilibrium.*

Proof. We proceed by contradiction. Suppose $s = (s_1, s_2, \ldots, s_n)$ is locally optimal among $\mathcal{P}$-invariant strategy profiles that is not a Nash equilibrium. We will construct an $s'$ arbitrarily close to $s$ with $EU(s') > EU(s)$.

Without loss of generality, suppose $s_1$ is not a best response to $s_{-1}$ but that the pure strategy of always playing $a_1$ is a best response to $s_{-1}$. For an arbitrary probability $p > 0$, consider the modified strategy $s_1'$ that plays action $a_1$ with probability $p$ and follows $s_1$ with probability $1-p$. Now, construct $s' = (s_1', s_2', \ldots, s_n')$ as follows:

$$s_i' = \begin{cases} s_i' = s_1' & \text{if } i \in \mathcal{P}(1) \\ s_i' = s_i & \text{otherwise.} \end{cases}$$

In words, $s'$ modifies $s$ by having the members of player 1's orbit mix in a probability $p$ of playing $a_1$. We claim for all sufficiently small $p$ that $EU(s') > EU(s)$.

To establish this claim, we break up the expected utility of $s'$ according to cases of how many players in 1's orbit play the action $a_1$ because of mixing in $a_1$ with probability $p$. In particular, we observe

$$\begin{aligned} EU(s') = &B(m{=}0, p)EU(s) \\ &+ B(m{=}1, p)EU((s_1', s_2, \ldots, s_n)) \\ &+ B(m{>}1, p)EU(\ldots), \end{aligned}$$

where $B(m, p)$ is the probability of $m$ successes for a binomial random variable on $m$ independent events that each have success probability $p$ and where $EU(\ldots)$ is arbitrary. Note that the crucial step in writing this expression is grouping the terms with the coefficient $B(m{=}1, p)$. We can do this because for any player $j \in \mathcal{P}(1)$, there exists a symmetry $\rho \in \Gamma(\mathcal{G})$ with $\rho(j) = 1$.

Now, to achieve $EU(s') > EU(s)$, we require

$$\begin{aligned} EU(s) < &\frac{B(m=1,p)}{B(m>0,p)} EU((s_1', s_2, \ldots, s_n)) \\ &+ \frac{B(m>1,p)}{B(m>0,p)} EU(\ldots). \end{aligned}$$

We know $EU((s_1', s_2, \ldots, s_n)) > EU(s)$, but we must deal with the case when $EU(\ldots)$ is arbitrarily negative. Because $\lim_{p\to 0} B(m > 1, p)/B(m = 1, p) = 0$, by making $p$ sufficiently small, $B(m = 1, p)/B(m > 0, p)$ can be made greater than $B(m > 1, p)/B(m > 0, p)$ by an arbitrarily large ratio. The result follows. ◻

## B OPTIMAL SYMMETRIC POLICY FOR THE RADIO STATION GAME OF EXAMPLE 3.3

We here calculate the optimal $\Gamma(\mathcal{G})$-invariant strategy profile for Example 3.3. Let $p$ be the probability of broadcasting the weather forecasts. By symmetry of the game and linearity of expectation, the expected utility given $p$ is simply four times the expected utility of any individual neighborhood. The value of an individual neighborhood is 0 with probability $(1-p)^3$, is 1 with probability $p^3$ and is 2 with the remaining probability. Hence, the expected utility of a single neighborhood is

$$p^3 + (1 - (1-p)^3 - p^3) \cdot 2 = 2 - 2(1-p)^3 - p^3.$$

The maximum of this term (and thus the maximum of the overall utility of all neighborhoods) can be found by any computer algebra system to be $p = 2 - \sqrt{2}$, which gives an expected utility of $4(\sqrt{2} - 1) \approx 1.66$.

To double-check, we can also calculate the symmetric Nash equilibrium of this game. It's easy to see that that Nash equilibrium must be mixed and therefore must make each player (radio station) indifferent about what to broadcast. So let $p$ again be the probability with which everyone else broadcasts the weather. The expected utility of broadcasting the weather relative to broadcasting nothing for any of the three relevant neighborhoods is $2(1-p)^2$. (Broadcasting the weather lifts the utility of a neighborhood from 0 to 2 if they do not already get the weather. Otherwise, it is useless to air the weather.) The expected utility of broadcasting music again relative to broadcasting nothing is simply $p^2$. We can find the symmetric Nash equilibrium by setting

$$2(1-p)^2 = p^2,$$

which gives us the same solution for $p$ as before.

## C PROOFS OF SECTION 4 RESULTS

Proposition 4.2. *Let $\mathcal{G}$ be a common-payoff normal-form game, and let $s^*$ be a locally-optimal $\mathcal{P}$-invariant strategy profile for some subset of game symmetries $\mathcal{P} \subseteq \Gamma(\mathcal{G})$. Suppose $G'$ is an $\epsilon$-perturbation of $\mathcal{G}$. Then $s^*$ is a $2\epsilon$-Nash equilibrium in $\mathcal{G}'$.*

Proof. By Theorem 3.2, $s^*$ is a Nash equilibrium in $\mathcal{G}$. After perturbing $\mathcal{G}$ by $\epsilon$ to form $\mathcal{G}'$, payoffs have increased / decreased at most $\pm\epsilon$, so the difference between any two actions' expected payoffs has changed by at most $2\epsilon$. ◻

Theorem 4.3. *Let $\mathcal{G}$ be a common-payoff normal-form game, and let $s^*$ be a locally-optimal $\mathcal{P}$-invariant strategy profile for some subset of game symmetries $\mathcal{P} \subseteq \Gamma(\mathcal{G})$. Suppose $s$ is a strategy profile with total variation distance $TV(s, s^*) \leq \delta$. Then $s$ is an $\epsilon$-Nash equilibrium with $\epsilon = 4\delta \max_{i\in N, a\in A} |u_i(a)|$.*

Proof. Consider the perspective of an arbitrary player $i$. The difference in expected utility of playing any action $a_i$ between the opponent strategy profiles $s_{-i}$ and $s_{-i}^*$ is given by:

$$\begin{aligned} &\left|EU_i(a_i, s_{-i}) - EU_i(a_i, s_{-i}^*)\right| \\ &= \left| \sum_{a_{-i}\in A_{-i}} s_{-i}(a_{-i}) u_i(a_i, a_{-i}) \right. \\ &\quad \left. - \sum_{a_{-i}\in A_{-i}} s_{-i}^*(a_{-i}) u_i(a_i, a_{-i}) \right| \\ &\leq \sum_{a_{-i}\in A_{-i}} |u_i(a_i, a_{-i})| \left|s_{-i}(a_{-i}) - s_{-i}^*(a_{-i})\right| \\ &\leq 2TV(s, s^*) \max_{i\in N, a\in A} |u_i(a)| \\ &\leq 2\delta \max_{i\in N, a\in A} |u_i(a)|. \end{aligned}$$

In particular, let $a_i$ be an action in the support of $s_i^*$, and let $a_i'$ be any other action. Then, using the above, we have:

$$\begin{aligned}
& EU_i(a_i', s_{-i}) - EU_i(a_i, s_{-i}) \\
&= EU_i(a_i', s_{-i}) - EU_i(a_i', s_{-i}^*) + EU_i(a_i', s_{-i}^*) \\
&\quad - EU_i(a_i, s_{-i}^*) + EU_i(a_i, s_{-i}^*) - EU_i(a_i, s_{-i}) \\
&\leq EU_i(a_i', s_{-i}) - EU_i(a_i', s_{-i}^*) \\
&\quad + EU_i(a_i, s_{-i}^*) - EU_i(a_i, s_{-i}) \\
&\leq \left|EU_i(a_i', s_{-i}) - EU_i(a_i', s_{-i}^*)\right| \\
&\quad + \left|EU_i(a_i, s_{-i}) - EU_i(a_i, s_{-i}^*)\right| \\
&\leq 4\delta \max_{i\in N, a\in A} |u_i(a)|,
\end{aligned}$$

where $EU_i(a_i', s_{-i}^*) - EU_i(a_i, s_{-i}^*) \leq 0$ because $s_i^*$ is a Nash equilibrium by Theorem 3.2. □

Corollary 4.4. *Let $\mathcal{G}$ be a common-payoff normal-form game, and let $s^*$ be a locally-optimal $\mathcal{P}$-invariant strategy profile for some subset of game symmetries $\mathcal{P} \subseteq \Gamma(\mathcal{G})$. Suppose $s$ is a strategy profile with Kullback-Leibler divergence satisfying $D_{KL}(s||s^*) \leq \nu$ or $D_{KL}(s^*||s) \leq \nu$. Then $s$ is an $\epsilon$-Nash equilibrium with $\epsilon = 2\sqrt{2\nu}\max_{i\in N, a\in A}|u_i(a)|$.*

Proof. By Pinsker's inequality [35], we have

$$TV(s, s^*) \leq \sqrt{\frac{1}{2} D_{KL}(s||s^*)}.$$

As $TV(s, s^*) = TV(s^*, s)$ and with a similar application of Pinsker's inequality, we have by assumption that $TV(s, s^*) \leq \sqrt{\nu/2}$. Applying Theorem 4.3 with $\delta = \sqrt{\nu/2}$ yields the result. □

# D PROOFS OF SECTION 5 RESULTS

Theorem 5.5. *Let $\mathcal{G}$ be a non-degenerate common-payoff normal-form game, and let $\mathcal{P} \subseteq \Gamma(\mathcal{G})$ be a subset of the game symmetries $\Gamma(\mathcal{G})$. A locally optimal $\mathcal{P}$-invariant strategy profile is locally optimal among possibly-asymmetric strategy profiles if and only if it is deterministic.*

Proof. Let $s$ be a locally optimal $\mathcal{P}$-invariant strategy profile. By Theorem 3.2, $s$ is a Nash equilibrium. Because $\mathcal{G}$ is non-degenerate, so is $s$. We prove the claim by proving that (1) if $s$ is deterministic, it is locally optimal in asymmetric strategy space; and (2) if $s$ is mixed then it is not locally optimal in asymmetric strategy space.

(1) The deterministic case: Let $s$ be deterministic. Now consider a potentially asymmetric strategy profile $s'$. We must show as $s'$ becomes sufficiently close to $s$ that $EU(s') \leq EU(s)$.

Let $\epsilon_1, \epsilon_2, ..., \epsilon_n$ and $\hat{s}_1, ..., \hat{s}_n$ be such that for $i \in N$, $s_i'$ can be interpreted as following $s_i$ with probability $1 - \epsilon_i$ and following $\hat{s}_i$ with probability $\epsilon_i$, where $s_i \notin \text{supp}(\hat{s}_i)$. Then (similar to the proof of Theorem 3.2), we can write

$$\begin{aligned}
& EU(s') \\
&= \left(\prod_{i\in N}(1-\epsilon_i)\right) EU(s) \\
&+ \sum_{j\in N} \epsilon_j \left(\prod_{i\in N-\{j\}} 1-\epsilon_i\right) \cdot EU(\hat{s}_j, s_{-j}) \\
&+ \sum_{j,l\in N: j\neq l} \epsilon_j\epsilon_i \left(\prod_{i\in N-\{j,l\}} 1-\epsilon_i\right) \cdot EU(\hat{s}_j, \hat{s}_l, s_{-j-l}) \\
&+ \ldots
\end{aligned}$$

The second line is the expected value if everyone plays $s$, the third line is the sum over the possibilities of one player $j$ deviating to $\hat{s}_j$, and so forth. We now make two observations. First, because $s$ is a Nash equilibrium, the expected utilities $EU(\hat{s}_j, s_{-j})$ in the third line are all at most as big as $EU(s)$. Now consider any later term corresponding to the deviation of some set $C$, containing at least two players $i, j$. Note that it may be $EU(\hat{s}_C, s_{-C}) > EU(s)$. However, this term is multiplied by $\epsilon_i\epsilon_j$. Thus, as the $\epsilon$ go to 0, the significance of this term in the average vanishes in comparison to that of both the terms corresponding to the deviation of just $i$ and just $j$, which are multiplied only by $\epsilon_i$ and $\epsilon_j$, respectively. By non-degeneracy, it is $EU(\hat{s}_i, s_{-i}) < EU(s)$ or $EU(\hat{s}_j, s_{-j}) < EU(s)$. Thus, if the $\epsilon_i$ are small enough, the overall sum is less than $EU(s)$.

(2) The mixed case: Let $s$ be mixed. We proceed by constructing a strategy profile $s'$ that is arbitrarily close to $s$ with $EU(s') > EU(s)$.

Let $m$ be the largest integer where for all subsets of players $C \subseteq N$ with $|C| \leq m$, the expected payoff is constant across all joint deviations to $a_i \in \text{supp}(s_i)$ for all $i \in C$, i.e., where $EU(a_C, s_{-C}) = EU(s)$ for all $a_C \in \text{supp}(s_C)$. As $s$ is a non-degenerate Nash equilibrium, $1 \leq m < n$.

By definition of $m$, there exists a subset of players $C \subset N$ with $|C| = m$ and choice of actions $a_C \in \text{supp}(s_C)$ where $EU(a_j, a_C, s_{-j-C})$ is not constant across the available actions $a_j \in A_j$ for some player $j \in N - C$. Denote player $j$'s best response to the joint deviation $a_C$ as $a_j^* \in \text{argmax}_{a_j} EU(a_j, a_C, s_{-j-C})$, and note $EU(a_j, a_C, s_{-j-C}) > EU(a_C, s_{-C}) = EU(s)$.

To construct $s'$, modify $s$ by letting player $j$ mix according to $s_j$ with probability $(1 - \epsilon)$ and play action $a_j$ with probability $\epsilon$. Similarly, let each player $i \in C$ mix according to $s_i$ with probability $(1 - \epsilon)$ and play their action $a_i$ specified by $a_C$ with probability $\epsilon$. Because we allow $\epsilon > 0$ to be arbitrarily small, all we have left to show is $EU(s') > EU(s)$.

Observe as before that we can break $EU(s')$ up into cases based on the number of players who deviate according to the modified

| | | Player 2 | |
|---|---|---|---|
| | | $\alpha$ | $\beta$ |
| Player 1 | $\alpha$ | $u_{\alpha\alpha}$ | $u_{\alpha\beta}$ |
| | $\beta$ | $u_{\alpha\beta}$ | $u_{\beta\beta}$ |

**Table 2: A payoff matrix with $|N| = 2$ and $A_1 = A_2 = \{\alpha, \beta\}$ to illustrate GAMUT games. In a RandomGame, $u_{\alpha\alpha}$, $u_{\alpha\beta}$, and $u_{\beta\beta}$ are i.i.d. draws from $Unif(-100, 100)$. In a CoordinationGame, $u_{\alpha\alpha}$ and $u_{\beta\beta}$ are i.i.d. draws from $Unif(0, 100)$ while $u_{\alpha\beta}$ is a draw from $Unif(-100, 0)$. In a CollaborationGame, $u_{\alpha\alpha} = u_{\beta\beta} = 100$, and $u_{\alpha\beta}$ is a draw from $Unif(-100, 99)$.**

probability $\epsilon$:

$$\begin{aligned} &EU(s') \\ &= \left( \prod_{k \in C \cup \{j\}} (1-\epsilon) \right) EU(s) \\ &+ \sum_{l \in C \cup \{j\}} \epsilon \left( \prod_{k \in C \cup \{j\} : k \neq l} 1-\epsilon \right) EU(a_l, s_{-l}) \\ &+ \ldots \\ &+ \left( \prod_{k \in C \cup \{j\}} \epsilon \right) EU(a_j, a_C, s_{-j-C}). \end{aligned}$$

By construction, every value in the expected value calculation $EU(s')$ is equal to $EU(s)$ except for the last value $EU(a_j, a_C, s_{-j-C})$, which is greater than $EU(s)$. We conclude $EU(s') > EU(s)$. □

# E GAMUT DETAILS AND ADDITIONAL EXPERIMENTS

## E.1 GAMUT games

In Section 6.1, we analyzed all three classes of symmetric GAMUT games: RandomGame, CoordinationGame, and CollaborationGame. Below, we give a formal definiton of these game classes:

*Definition E.1.* A *RandomGame* with $|N|$ players and $|A|$ actions assumes $A_i = A_j$ for all $i, j$ and draws a payoff from $Unif(-100, 100)$ for each unordered action profile $a \in A$.

*Definition E.2.* A *CoordinationGame* with $|N|$ players and $|A|$ actions assumes $A_i = A_j$ for all $i, j$. For each unordered action profile $a \in A$ with $a_i = a_j$ for all $i, j$, it draws a payoff from $Unif(0, 100)$; for all other unordered action profiles, it draws a payoff from $Unif(-100, 0)$.

*Definition E.3.* A *CollaborationGame* with $|N|$ players and $|A|$ actions assumes $A_i = A_j$ for all $i, j$. For each unordered action profile $a \in A$ with $a_i = a_j$ for all $i, j$, the payoff is 100; for all other unordered action profiles, it draws a payoff from $Unif(-100, 99)$.

Note that these games define payoffs for each *unordered* action profile because the games are totally symmetric (Definition 2.3). Table 2 gives illustrative examples.

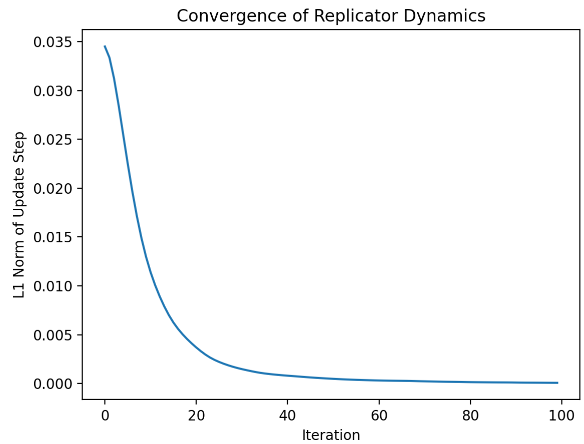


**Figure 2: The magnitude of the replicator dynamics update step averaged over 10,000 RandomGames[4] with 2 players and 2 actions. Although this plot indicates that the replicator dynamics converge by 100 iterations, we ran 10,000 iterations for good measure in all of our experiments.**

## E.2 Proof of Proposition 6.1

Proof. By Definition 5.4, in order for a game to be degenerate, there must exist a player $i$, a set of actions for the other players $a_{-i}$, and a pair of actions $a_i \neq a_i'$ with $EU(a_i, a_{-i}) = EU(a_i', a_{-i})$. In RandomGames, CoordinationGames, and CollaborationGames, $EU(a_i, a_{-i}) = \mu(a_i, a_{-i})$ and $EU(a_i', a_{-i}) = \mu(a_i', a_{-i})$ are continuous random variables that are independent of each other. (Or, in the case of a CollaborationGame, $\mu(a_i, a_{-i})$ may be a fixed value outside of the support of $\mu(a_i', a_{-i})$.) So $EU(a_i, a_{-i}) = EU(a_i', a_{-i})$ is a measure-zero event. □

## E.3 Replicator dynamics

Consider a game where all players share the same action set, i.e., with $A_i = A_j$ for all $i, j$, and consider a totally symmetric strategy profile $s = (s_1, s_1, \ldots, s_1)$. In the replicator dynamic, each action can be viewed as a species, and $s_1$ defines the distribution of each individual species (action) in the overall population (of actions). At each iteration of the replicator dynamic, the prevalence of an individual species (action) grows in proportion to its relative fitness in the overall population (of actions). In particular, the replicator dynamic evolves $s_1(a)$ over time $t$ for each $a \in A_1$ as follows:

$$\frac{d}{dt} s_1(a) = s_1(a) \left[ EU(a, s_{-1}) - EU(s) \right].$$

To simulate the replicator dynamic with Euler's method, we need to choose a stepsize and a total number of iterations. Experimentally, we found the fastest convergence with a stepsize of 1, and we found that 100 iterations sufficed for convergence; see Figure 2. For good measure, we ran 10,000 iterations of the replicator dynamic in all of our experiments.

[4]In this simulation only we rescaled the RandomGames so that each payoff is a draw from $Unif(0, 1)$.

We are interested in the replicator dynamic for two reasons. First, it is a model for how agents in the real world may collectively arrive at a symmetric solution to a game (e.g., through evolutionary pressure). Second, it is a learning algorithm that performs local search in the space of symmetric strategies. In our experiments of Appendix E.5, we find that using the replicator dynamic as an optimization algorithm is competitive with Sequential Least SQuares Programming (SLSQP), a local search method from the constrained optimization literature [17, 40].

## E.4 What fraction of symmetric optima are local optima in possibly-asymmetric strategy space?

As discussed in Section 6.2, we would like to get a sense for how often symmetric optima are stable in the sense that they are also local optima in possibly-asymmetric strategy space (see Section 5). Table 3 shows in what fraction of games the best solution we found is *unstable*.

## E.5 How often do SLSQP and the replicator dynamic find an optimal solution?

As discussed in Section 6.3, Table 4 and Table 5 show how often SLSQP finds an optimal solution, while Table 6 and Table 7 show how often the replicator dynamic finds an optimal solution.

## E.6 How costly is payoff perturbation under the simultaneous best response dynamic?

When a game is not stable in the sense of Section 5, we would like to understand how costly the worst-case $\epsilon$-perturbation of the game can be. (See Definition 4.1 for the definition of an $\epsilon$-perturbation of a game.) In particular, we study the case when individuals simultaneously update their strategies in possibly-asymmetric ways by defining the following *simultaneous best response dynamic*:

*Definition E.4.* The *simultaneous best response dynamic at s* updates from strategy profile $s = (s_1, s_2, \ldots, s_n)$ to strategy profile $s' = (s'_1, s'_2, \ldots, s'_n)$ with every $s'_i$ a best response to $s_{-i}$.

For each of the RandomGames in Section 6.2 whose symmetric optimum $s$ is not a local optimum in possibly-asymmetric strategy space, we compute the worst-case $\epsilon$ payoff perturbation for infinitesimal $\epsilon$. Then, we update each player's strategy according to the simultaneous best response dynamic at $s$. This necessarily leads to a decrease in the original common payoff because the players take simultaneous updates on an objective that, after payoff perturbation, is no longer common. Table 8 reports the average percentage decrease in expected utility, which ranges from 55% to 89%. Our results indicate that simultaneous best responses after payoff perturbation in RandomGames can be quite costly.

# F THE COMPUTATIONAL COMPLEXITY OF FINDING THE SYMMETRIES OF A GAME

In this section, we analyze the computational complexity of finding the symmetries of a common-payoff game. In general, symmetries as defined in Definition 2.1 can be found in exponential time in the number of players. Therefore, if we represent the game explicitly as a full payoff matrix, then the symmetries can be found in polynomial time in the size of the input. However, if we can represent the game more efficiently by giving only non-zero entries of the payoff matrix, the problem becomes graph isomorphism-complete, i.e., polynomial-time equivalent to the graph isomorphism problem, which is neither known to be solvable in polynomial time nor known to be NP-hard [see 11, for an overview]. We also show (in Section F.3) that if we consider a more general notion of game symmetry that permutes actions in addition to players, the computational problem becomes graph isomorphism-complete on an explicit payoff matrix representation.

## F.1 The hypergraph automorphism problem

We here introduce the hypergraph automorphism problem and some existing results about it. In the next section, we will prove our results by relating the game automorphism problem (i.e., the problem of finding the symmetries of a game) to the hypergraph automorphism problem.

A *hypergraph* is a pair $(V, E)$, where $V$ is a (finite) set of vertices and $E \subseteq 2^V$ is a set of *hyperedges*.

A *symmetry* or automorphism of a hypergraph is a bijection $\rho\colon V \to V$ s.t. for each set of vertices $e \subseteq V$, it is $e \in E$ if and only if $\rho(e) \in E$, where $\rho(e) := \{\rho(v) \mid v \in e\}$. In other words: For $\rho$ to be a symmetry it must be the case that any set of vertices $v_1, \ldots, v_k$ are connected by a hyperedge if and only if $\rho(v_1), \ldots, \rho(v_k)$ are connected by a hyperedge.

*Definition F.1.* The *hypergraph automorphism (HA) problem* asks for a given hypergraph $(V, E)$ to provide a set of symmetries of $(V, E)$ that generate the group of all symmetries of $(V, E)$.

There are two natural ways to represent the edges of a hypergraph. The first is to provide what one would call an adjacency matrix in the case of a regular graph. That is, we give a table of bits that specifies for each $e \in 2^V$ whether $e \in E$. That is, for each set of vertices, we specify whether there is a hyperedge connecting that set of vertices. The downside of this representation style is that it always costs $O(2^{|V|})$ bits. An alternative is to explicitly list $E$, such that graphs with few edges can be represented in much less space than $O(2^{|V|})$. We call the former notation *non-sparse* and the latter *sparse*.

LEMMA F.2. *The HA problem on a sparse hypergraph representation is graph isomorphism-complete.*

PROOF. Mathon [21] shows that the problem of giving the generators of the *auto*morphism group of a given graph *iso*morphism complete. So it is left to show that the automorphism problem on sparse hypergraphs is polynomially equivalent to analogous problem on graphs. Since graphs are hypergraphs, we only need to reduce HA to the graph automorphism problem. This is easy and the main idea has been noted before, e.g., see the introduction of Arvind et al. [1]. □

THEOREM F.3 (LUKS, 1999, THEOREM 4.2). *The HA problem is solvable in $O(c^{|V|})$ for some constant $c$. In particular, it follows immediately that HA on non-sparse representations is solvable in polynomial time.*

| A<br>N | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 2 | 0.36 | 0.44 | 0.44 | 0.50 |
| 3 | 0.38 | 0.49 | 0.59 | 0.60 |
| 4 | 0.42 | 0.45 | 0.46 | 0.46 |
| 5 | 0.45 | 0.48 | 0.49 | 0.47 |

**(a) RandomGame**

| A<br>N | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 |

**(b) CoordinationGame**

**Table 3: The fraction of games whose symmetric optima are mixed. By Theorem 5.5, these symmetric equilibria are the ones *unstable* in the sense of Section 5. Numbers in the table were empirically determined from 100 randomly sampled games per GAMUT class.**

| A<br>N | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 2 | 0.92 | 0.81 | 0.70 | 0.64 |
| 3 | 0.80 | 0.69 | 0.57 | 0.48 |
| 4 | 0.75 | 0.57 | 0.40 | 0.35 |
| 5 | 0.70 | 0.45 | 0.36 | 0.31 |

**(a) RandomGame**

| A<br>N | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 2 | 0.59 | 0.50 | 0.40 | 0.33 |
| 3 | 0.53 | 0.38 | 0.28 | 0.29 |
| 4 | 0.53 | 0.37 | 0.29 | 0.26 |
| 5 | 0.53 | 0.36 | 0.33 | 0.25 |

**(b) CoordinationGame**

**Table 4: The fraction of single SLSQP runs that achieve the best solution found in our 20 total optimization attempts. Numbers in the table were empirically determined from 100 randomly sampled games per GAMUT class.**

| A<br>N | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 2 | 1.00 | 0.99 | 0.99 | 0.98 |
| 3 | 1.00 | 0.99 | 1.00 | 0.96 |
| 4 | 1.00 | 0.96 | 0.94 | 0.88 |
| 5 | 0.98 | 0.90 | 0.88 | 0.91 |

**(a) RandomGame**

| A<br>N | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 2 | 0.99 | 1.00 | 0.98 | 0.97 |
| 3 | 1.00 | 0.99 | 0.93 | 0.95 |
| 4 | 1.00 | 0.97 | 0.97 | 0.93 |
| 5 | 0.99 | 1.00 | 0.95 | 0.92 |

**(b) CoordinationGame**

**Table 5: The fraction of games in which at least 1 of 10 SLSQP runs achieves the best solution found in our 20 total optimization attempts. Numbers in the table were empirically determined from 100 randomly sampled games per GAMUT class.**

| A<br>N | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 2 | 0.93 | 0.81 | 0.68 | 0.65 |
| 3 | 0.81 | 0.70 | 0.58 | 0.46 |
| 4 | 0.76 | 0.58 | 0.36 | 0.34 |
| 5 | 0.69 | 0.43 | 0.36 | 0.30 |

**(a) RandomGame**

| A<br>N | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 2 | 0.58 | 0.45 | 0.40 | 0.33 |
| 3 | 0.57 | 0.35 | 0.29 | 0.27 |
| 4 | 0.53 | 0.37 | 0.28 | 0.25 |
| 5 | 0.51 | 0.33 | 0.33 | 0.24 |

**(b) CoordinationGame**

**Table 6: The fraction of single replicator dynamics runs that achieve the best solution found in our 20 total optimization attempts. Numbers in the table were empirically determined from 100 randomly sampled games per GAMUT class.**

This is harder to show. Note that a brute force method that tests all $|V|!$ bijections is super-exponential in $|V|$ and super-polynomial in (the problem size) $2^{|V|}$.

| A N | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 2 | 1.00 | 1.00 | 1.00 | 1.00 |
| 3 | 0.99 | 1.00 | 0.95 | 0.96 |
| 4 | 1.00 | 0.98 | 0.91 | 0.91 |
| 5 | 0.98 | 0.97 | 0.92 | 0.87 |

**(a) RandomGame**

| A N | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 2 | 1.00 | 1.00 | 0.99 | 0.94 |
| 3 | 1.00 | 0.97 | 0.93 | 0.96 |
| 4 | 0.99 | 1.00 | 0.93 | 0.92 |
| 5 | 1.00 | 0.98 | 0.96 | 0.90 |

**(b) CoordinationGame**

**Table 7: The fraction games in which at least 1 of 10 replicator dynamics runs achieves the best solution found in our 20 total optimization attempts. Numbers in the table were empirically determined from 100 randomly sampled games per GAMUT class.**

| A N | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 2 | 58.9% | 55.9% | 61.8% | 64.6% |
| 3 | 73.7% | 70.9% | 73.4% | 73.7% |
| 4 | 74.1% | 77.4% | 78.4% | 82.5% |
| 5 | 77.4% | 84.9% | 89.9% | 87.5% |

**(a) RandomGame**

**Table 8: The average decrease in expected utility that worst-case infinitesimal asymmetric payoff perturbations cause to unstable symmetric optima. To get these numbers, we first perturb payoffs in the 100 RandomGames from Section 6.2 whose symmetric optima $s$ are not local optima in possibly-asymmetric strategy space. Then, in each perturbed game, we compute a simultaneous best-response update to $s$ and record its decrease in expected utility.**

## F.2 Polynomial-time equivalence

Recall from the main text that – for the purpose of our paper – a symmetry of an $n$-player (common-payoff) game $(A, u)$ is a permutation $\rho\colon \{1, ..., n\} \rightarrow \{1, ..., n\}$ s.t. for all pure strategy profiles $\mathbf{a} \in A$, it is $u(a_1, ..., a_n) = u(a_{\rho(1)}, ..., a_{\rho(n)})$. In particular, we do not consider permutations of the actions.

*Definition F.4.* The *(common-payoff) game automorphism (GA) problem* asks us to compute for a common-payoff given game a generating set of the symmetries of the game.

As with hypergraphs, we distinguish two representations for a game. A sparse representation lists only non-zero payoffs. A non-sparse representation gives the payoff for each $A$. As before, the downside of the full payoff table representation is that its size is exponential in the number of players.

THEOREM F.5. *The sparse/non-sparse representation HA problem is polynomial-time equivalent to the sparse/non-sparse representation GA problem.*

By polynomial-time, we here mean in time bound by a polynomial in the number of players and the size in bits of the given instance.

PROOF. HA→GA: We first reduce the HA problem to the GA problem, which is the easier direction. We will use the same construction for both the sparse-to-sparse and non-sparse-to-non-sparse case.

Take a given hypergraph $(V, E)$. WLOG assume $V = \{1, ..., n\}$. Then we construct an $n$-player game, in which each player has two actions, $a_0, a_1$. For any $M \subseteq \{1, ..., n\}$, let $\mathbf{a}_M$ be the payoff profile in which the set of players playing $a_1$ is exactly $M$. Then let the payoff $u(\mathbf{a}_M)$ be 1 if $M \in E$ and 0 otherwise.

We now show that this reduction is valid by showing that the game and the graph have the same symmetries. Let $\rho$ be a bijection. Then:

$$\begin{aligned} & \rho \text{ is a symmetry of } (V, E) && (1) \\ \text{iff} \quad & \forall e \in 2^V : e \in E \iff \rho(e) \in E && (2) \\ \text{iff} \quad & \forall M \in 2^V : u(\mathbf{a}_M) = 1 \iff u(\mathbf{a}_{\rho(M)}) = 1 && (3) \\ \text{iff} \quad & \forall M \in 2^V : u(\mathbf{a}_M) = u(\mathbf{a}_{\rho(M)}) && (4) \\ \text{iff} \quad & \forall \mathbf{a} \in A\colon u(a_1, ..., a_n) = u(a_{\rho(1)}, ..., a_{\rho(n)}) && (5) \\ \text{iff} \quad & \rho \text{ is a symmetry of } (A, u) && (6) \end{aligned}$$

It is easy to see that this construction can be performed in polynomial (indeed linear) time for both sparse and non-sparse representations.

GA→HA: We now reduce in the opposite direction. This is more complicated and we therefore provide only a sketch.

Consider an $n$-player game $(A, u)$. We construct the hypergraph as follows. First, for each player $i$, we generate a vertex. We also generate $\log_2(|A_i|)$ vertices that we use to encode $A_i$, player $i$'s actions and connect them with the vertex representing $i$. For players $i, j$ that have the same action label sets, this encoding must be done consistently for $i, j$.

We also need to add some kind of structure to ensure that symmetries of the hypergraph can only map the $k$th action-encoding vertex of player $i$ on the $k$th action-encoding vertex of a player $j$ that has the same action label set as $i$.

Next, we represent the payoff function $u$. To do so, we introduce $\lceil \log_2(|u(A) - \{0\}|) \rceil$ payoff encoding vertices. Note that $|u(A) - \{0\}|$ is the number of distinct non-zero payoffs of the game. To encode $|u(A) - \{0\}|$ we therefore need $\lceil \log_2(|u(A) - \{0\}|) \rceil$ bits. We connect these bits in such a way that any symmetry must map each of them onto itself. We fix some binary encoding of $u(A) - \{0\}$. For instance, let's say the non-zero payoffs of the game are $\{-3, -1, 7, 8, 10, 11, 13, 100\}$. Then we need three bits, and might encode them as $-3 \mapsto 000$, $-1 \mapsto 001$, $7 \mapsto 010$, $8 \mapsto 011$, and so forth.

For each $\mathbf{a} \in A$ with $u(\mathbf{a}) \neq 0$, we then add an edge that contains for each Player $i$ the action encoding vertices corresponding to $a_i$; and those bits from the payoff encoding vertices that together represent the payoff. (So, for example, if the payoff is encoded by 011, then the hyperedge contains the two lower payoff encoding vertices. Similarly for the action encoding vertices.)

We omit a proof of the correctness of this reduction.

It is left to show that the reduction is polynomial-time for both representation styles. For the sparse representation styles, it is trivial because up to some small number of extra vertices and edges, there is a one-to-one correspondence between edges of the hypergraph and action profiles with non-zero payoffs.

On to the non-sparse representation. Clearly each entry of the adjacency matrix can be filled in polynomial (perhaps even constant or logarithmic) time. It is left to show that the adjacency matrix is not too large. In particular, we need to show that the size of the adjacency matrix is polynomial in the size of the payoff matrix. To assess the size of the adjacency matrix we need to count the number of vertices in the above construction. First, the number of player vertices is

$$\begin{aligned} n &\leq \log_2(|A_1|) + ... + \log_2(|A_n|) \\ &= \log_2(|A_1| \cdot ... \cdot |A_n|) \\ &= \log_2(|A|). \end{aligned}$$

(The inequality assumes each player has at least two actions.) Second, the number of action-encoding vertices is

$$\begin{aligned} &\lceil \log_2(|A_1|) \rceil + ... + \lceil \log_2(|A_n|) \rceil \\ \leq\ & 2\log_2(|A_1|) + ... + 2\log_2(|A_n|) \\ =\ & 2\log_2(|A|). \end{aligned}$$

Finally, the number of payoff encoding vertices is about

$$\lceil \log_2(|u(A)|) \rceil \leq 2\log_2(|u(A)|) \leq 2\log_2(|A|).$$

The overall number of vertices in the above construction is therefore at most $5\log_2(|A|)$. Thus, the size (in terms of number of bits) of the adjacency matrix is bound by

$$2^{5(\log_2 |A|)} = |A|^5.$$

Since $|A|$ is a lower bound on the size of the payoff matrix (in bits), this is polynomial in the size of the game's payoff matrix, as required. □

One might wonder: In the non-sparse representation case, why does the reduction to HA not also work if we use a more traditional sense of game symmetries? If it were to work that would show that GI is polynomial-time solvable! But this does not work (with the proof strategy used above). In the current reduction, actions do not get their own vertices. Thus, (even if we dropped the constraint structures that prevent actions from being remapped), a hypergraph automorphism cannot remap, e.g., an action encoded as 11011 to an action encoded as 01010. To express full action relabelings in the hypergraph, it seems that we need to introduce an action per vertex. However, the size of the adjacency matrix then blows up more than polynomially.

Combining Lemma F.2 and Theorems F.3 and F.5, we get a characterization of the complexity of the graph automorphism problem.

Corollary F.6. *GA is solvable in polynomial time on a non-sparse representation and is GI-complete on a sparse representation.*

## F.3 An alternative notion of game symmetry

As mentioned in the main text, we only consider symmetries that relabel the players and the above is on the computational problem resulting from that notion of symmetry. As noted in footnote 3, this was done in part to keep notation simple and an alternative, slightly more complicated notion allows for the actions to be permuted. A natural question then is what the complexity is of finding this new type of symmetry in a given common-payoff game. In this case, the answer is that finding symmetries is GI-complete regardless of how the game is represented and follows almost immediately from existing ideas from Mathon [21] and Gabarró et al. [10].

A player-action (PA) symmetry of a game is pair of a bijection $\rho\colon \{1, ..., n\} \to \{1, ..., n\}$ on players and a family of bijections $\tau_i\colon A_{\rho(i)} \to A_i$ s.t. for all pure strategy profiles $(a_1, ..., a_n)$,

$$u(a_1, ..., a_n) = u(\tau_1(a_{\rho(1)}), ..., \tau_n(a_{\rho(n)})).$$

So the idea in this new definition is that $\tau_i$ translates the action names from those of player $\rho(i)$ to player $i$.

Define the PAGA problem analogously to the GA problem above, as finding a generating set of the PA symmetries of a given common-payoff game. This time, the complexity is independent of whether the game is represented sparsely or not.

Theorem F.7. *The PAGA problem is GI-complete.*

Proof. PAGA→GI For their proof of GI-completeness of the game isomorphism problem, Gabarró et al. [10] sketch how a general-sum game can be represented as a graph in a way that maintains the isomorphisms. In particular, we can therefore represent a single common-payoff game as a graph in a way that maintains PA symmetries. The have thus given a sketch of a polynomial-time reduction from PAGA to the problem of finding the automorphisms of a graph. This latter problem can in turn be reduced in polynomial time to the graph isomorphism problem as was shown by Mathon [21].

GI→PAGA Second, we show GI-hardness. As shown by Mathon [21], it is enough to reduce the graph automorphism problem to the PAGA problem. For this, we can slightly modify the construction of Gabarró et al. [10, Lemma 5]. Specifically, they reduce the graph *iso*morphism problem to the 4-player *general-sum* game *iso*morphism problem, where actions represent vertices and the graph isomorphisms can be recovered from those PA game isomorphisms where the player isomorphism is $\rho = \mathrm{id}$. Obviously, the same construction can be used to reduce the graph *auto*morphism problem to the general-sum game *auto*morphism problem. The only issue is therefore that their construction uses general-sum games, but we can simply encode their payoff vectors as single numbers. In particular, because their payoffs are binary, we might translate $(0,0,0,0) \mapsto 0$, $(0,0,0,1) \mapsto 1$, $(0,0,1,0) \mapsto 2$, and so forth. It's easy to see that the symmetries with $\rho = \mathrm{id}$ remain the same under this transformation. □

# G THE COMPUTATIONAL COMPLEXITY OF FINDING OPTIMAL SYMMETRIC STRATEGIES

## G.1 Polynomials

A (multivariate) polynomial in $k$ variables is a function

$$(x_1, ..., x_k) \mapsto \sum_{e_1,...,e_k \in \{1,...,m\}} c_{(e_1,...,e_k)} x_1^{e_1} ... x_1^{e_k}$$

for some $m$, where the $c_{(e_1,...,e_k)}$ are some set of real coefficients. The terms $c_{(e_1,...,e_k)} x_1^{e_1} ... x_1^{e_k}$ for which $c_{(e_1,...,e_k)} \neq 0$ are called the monomials of the polynomial. The *maxdegree of a monomial* $c_{(e_1,...,e_k)} x_1^{e_1} ... x_1^{e_k}$ is $\max_{i=1,...,k} e_i$. The *maxdegree of a polynomial* is the maximum of the maxdegrees of its monomials. Similarly, the *total degree of a monomial* $c_{(e_1,...,e_k)} x_1^{e_1} ... x_1^{e_k}$ is $\sum_{i=1}^k e_i$. The *total degree of a polynomial* is the maximum of the total degrees of its monomials. The degree of a variable $x_i$ in a monomial $c_{(e_1,...,e_k)} x_1^{e_1} ... x_1^{e_k}$ is $e_i$. The maxdegree of $x_i$ in a polynomial is the maximum of the maxdegrees of $x_i$ in all of the monomials.

We can partition the parameters of a polynomial into vectors and write the polynomial as $f(\mathbf{x}_1, ..., \mathbf{x}_k)$, where $\mathbf{x}_1, ..., \mathbf{x}_k$ are real vectors. We define the degree of $\mathbf{x}_i$ in a monomial as the sum of the degrees of the entries of $\mathbf{x}_i$ in the monomial. We define the maxdegree of $\mathbf{x}_i$ in the polynomial as the maximum of the degrees of $\mathbf{x}_i$ in the polynomial's monomials.

In the following we will interpret the set $\Delta(A_i)$ of probability distributions over $A_i$ as the set of $|A_i|$-dimensional vectors of non-negative reals whose entries sum to one. We will index these vectors by $A_i$ (rather than numbers $1, ..., |A_i|$). The sets $\Delta(A_i)$ are also called unit simplices.

## G.2 Optimizing symmetric strategies as maximizing polynomials

It is immediately obvious that in a symmetric game, the expected utility as a function of the probabilities that each of the orbits assign to each of the strategies is a polynomial over a Cartesian product of unit simplices. Formally:

Proposition G.1. *Let $\mathcal{G}$ be an $n$-player game and $\mathcal{P} \subseteq \Gamma(\mathcal{G})$ be a subset of the game symmetries of $\mathcal{G}$. Let the orbits of $\mathcal{P}$ be $M_1, ..., M_k$. Further, let the set of actions of orbit $i$ be $A_i$. Then the expected utility function over $\mathcal{P}$-invariant strategy profiles of $\mathcal{G}$ is a polynomial over $\Delta(A_1) \times ... \times \Delta(A_k)$ with a max degree of (at most) $\max_i |M_i|$ and a total degree of (at most) $n$. This polynomial can be created in polynomial-time in the size of a sparse or non-sparse (as per Appendix F.2) representation of the game.*

It follows that we can use algorithms for optimizing polynomials to find optimal symmetric strategies and that positive results on optimizing polynomials transfer to finding optimal mixed strategies. Unfortunately, these results are generally somewhat cumbersome to state. This is because the optimum can in general not be represented exactly algebraically, even using $n$-th roots, as implied by the Abel–Ruffini theorem. Positive results must therefore be given in terms of approximations of the optimal solution. One striking result from the literature is that, roughly speaking, for a fixed number of variables, the optimal solution can be approximated in polynomial time [16, Section 6.1]. Translated to our setting, this means that the optimal symmetric strategy can be approximated in polynomial time if we keep constant the number of orbits and the number of actions available to each orbit, but potentially increase the number of players in each orbit. For more discussion of the complexity of optimizing polynomials on unit simplices, see de Klerk [6].

## G.3 Expressing polynomials as symmetric games

We now show that, conversely, for any polynomial over a Cartesian product of simplices there exists a symmetric game whose expected utility term is exactly that polynomial. However, depending on how we represent polynomials and how we represent games, the size of the game may blow up exponentially.

We first show that each polynomial over $\Delta(A_1) \times ... \times \Delta(A_k)$ can be rewritten in such a way that each input $\mathbf{x}_i$ appears in the same degree in all monomials.

Lemma G.2. *Let $f(\mathbf{x}_1, ..., \mathbf{x}_k)$ be a polynomial on real vectors of dimensions $A_1, ..., A_k$. Then there exists a polynomial $g$ on the same inputs s.t. for all $(\mathbf{x}_1, ..., \mathbf{x}_k) \in \Delta(A_1) \times ... \times \Delta(A_k)$*

$$g(\mathbf{x}_1, ..., \mathbf{x}_k) = f(\mathbf{x}_1, ..., \mathbf{x}_k),$$

*and the degree of every $\mathbf{x}_i$ in all monomials of $g$ is the maxdegree of $\mathbf{x}_i$ in $f$.*

Proof. Consider any monomial $\tilde{f}$ of $f$ in which $\mathbf{x}_i$ does not have its max degree. Then for all $(\mathbf{x}_1, ..., \mathbf{x}_k) \in \Delta(A_1) \times ... \times \Delta(A_k)$,

$$\begin{aligned} &\tilde{f}(\mathbf{x}_1, ..., \mathbf{x}_k) \\ = &\left( \sum_{a_i \in A_i} x_{i,a_i} \right) \tilde{f}(\mathbf{x}_1, ..., \mathbf{x}_k) \\ = &\sum_{a_i \in A_i} x_{i,a_i} \tilde{f}(\mathbf{x}_1, ..., \mathbf{x}_k). \end{aligned}$$

Notice that this is the sum of $|A_i|$ monomials in which $\mathbf{x}_i$ occurs in 1 plus the degree in which it occurs in $\tilde{f}$. We can iterate this transformation until we arrive at the desired $\tilde{f}$. □

Note, however, that if web take a given polynomial represented as a sum of monomials – e.g., $f(x_1, x_2) = x_1^4 - 3x_2$ – and rewriting it as outlined in the Lemma and its proof, the size may blow up exponentially. E.g., $f(x_1, x_2) = x_1^4 - 3x_2 = x_1^4 - 3(x_1 + x_2)^3 x_2$ and $(x_1 + x_2)^3$ expands into a sum of $2^3 = 8$ terms. However, in some table-of-coefficient representations of polynomials the size of the instance does not change at all and the transformation can be performed in polynomial time in the input. For example, this is the case if $k = 1$ and we represent a polynomial as a table of the coefficients of all terms $x_1^{e_1} ... x_k^{e_k}$ where $e_1 + ... + e_k$ are at most the polynomial's maxdegree.

Once we have a polynomial of the structure described in Lemma G.2, we can transform it into a game:

Proposition G.3. *Let $f(\mathbf{x}_1, ..., \mathbf{x}_k)$ be a polynomial in which each $\mathbf{x}_i$ appears in the same degree in all monomials. Then we can construct a game $\mathcal{G}$ with symmetries $\mathcal{P}$ that create $k$ orbits, where the number of players in orbit $i = 1, ..., k$ is the degree of $\mathbf{x}_i$ in $f$ and the number of actions for the players in orbit $i$ is the number of entries of $\mathbf{x}_i$.*

Proof. Consider games $\Gamma$ with orbits $M_1, ..., M_k$ of the specified sizes and sets of actions $A_1, ..., A_k$ also of the specified sizes where specifically the players in each $M_i$ are totally symmetric. Then such a game if fully specified as follows. For each family of numbers $n_{1,1}, ..., n_{1,|A_1|}, ...., n_{k,1}, ..., n_{k,|A_k|}$ with $n_{i,1} + ... + n_{i,|A_i|} = |M_i|$ we need to specify the utility $v$ obtained if for all $i, l$, $n_{i,j}$ players in orbit $i$ play action number $j$ from $A_i$.

In the expected utility function of $\mathcal{G}$, each such entry creates a summand

$$v \cdot \left( \prod_i \binom{|M_i|}{n_{i,1}, ..., n_{i,|A_i|}} \right) \prod_{i,l} p_{i,l}^{n_{i,l}},$$

where $p_{i,l}$ is the probability with which players in orbit $i$ player action $l$ and $\binom{|M_i|}{n_{i,1},...,n_{i,|A_i|}}$ is a multinomial. By setting $v$ appropriately, we can thus obtain any monomial with exponents $(n_{i,l})_{i,l}$. By setting the values $v$ all different sets of $(n_{i,l})_{i,l}$ appropriately, we obtain any polynomial in which each $(n_{i,l})_l$ appears with the same degree $|M_i|$ in all monomials. □

Note that if the polynomial is represented as a table of coefficients, then this reduction takes linear time in the size of the input. Similarly, if the polynomial is given as a list of only the monomials with non-zero coefficients – all of which satisfy the degree requirement – the reduction can also be done in polynomial time. This in particular gives us the following negative result, translated from the literature on optimizing polynomials:

Corollary G.4. *Deciding for a given game $\mathcal{G}$ with symmetries $\mathcal{P}$ and a given number $K$ whether there is a $\mathcal{P}$-invariant profile with expected utility at least $K$ is NP-hard, even for 2-player symmetric games.*

Proof. Follows from Proposition G.3 and the NP-hardness of optimizing quadratic polynomials over the unit simplex [6, Section 3.2]. □